TeamUCF - DARPA Urban Challenge

Technical Report

1 June 2007

University of Central Florida
College of Engineering & Computer Science • I^{2}Lab
School of Electrical Engineering & Computer Science
4000 Central Florida Blvd. • Orlando • Florida • 32816

Point of Contact:

Benjamin J. Patz
School of Electrical Engineering and Computer Science
UCF – 4000 Central Florida Blvd.
Orlando, FL 32816-2450
Phone: 407-963-5804
Email : bpatz@eecs.ucf.edu

Table of Contents

1. EXECUTIVE SUMMARY

TeamUCF and the Knight Rider vehicle were conceived over three years ago at the beginning of the 2005 DARPA Grand Challenge event. With the inception of the DARPA Urban Challenge, TeamUCF has built on the existing capabilities of the Knight Rider vehicle, creating a competition vehicle with the goal of meeting all the mission objectives of DARPA and performing at a level that challenges all other entries. TeamUCF's core university personnel have been augmented by an industry partner, Coleman Technologies, Inc., a system engineering firm specializing in real-time guidance, navigation, and control as well as products associated with GPS measurements in urban environments.

Our Urban Challenge vehicle is known as Knight Rider VIP and focuses on the implementation of higher-level capabilities through the use of Vision, Intelligence, and Planning in both a technical and motivational sense. Computer vision systems utilize a combination of sensors including active laser scanners, visible spectrum passive cameras, and near field acoustic sensors. UCF's renowned Computer Vision Lab is a key collaborator in this effort. Intelligence systems include a context-based reasoning subsystem which builds a real-time environmental model augmenting the supplied Route Network Definition File (RNDF) and develops mission plans to achieve objectives identified in the Mission Definition File (MDF). Planning provides tactical guidance information and vehicle control to achieve defined missions in a safe and effective manner.

The Knight Rider VIP has demonstrated its capabilities in numerous test scenarios at UCF's site visit demonstration facility, a flexible course for testing mission performance from parking trough in-traffic driving.

2. INTRODUCTION AND OVERVIEW

This report discusses design philosophies as well as engineering details associated with each of the key control system elements (Vision, Intelligence, and Planning). The following guidelines, coupled with requirements gleaned from DARPA Grand Challenge rules, scenario definitions, and FAQs were assumed as inputs to any design trades performed:

- Vehicle: UCF selected Subaru Outback Legacy
  - 4.8m overall length w/ mounting brackets
  - 2.0m overall width w/ mounting brackets
  - 2.6m wheelbase
  - 5.5m turning radius
- Performance constraints
  - Speed: -2.2 .. 13.5 m/s
  - Axial A: ~3.5 m/s^2
  - Lateral A: ~2 m/s^2
- General mission description
  - Autonomous navigation through a series of inertially-defined waypoints in a sparsely defined urban environment with a

Figure 1. Knight Rider VIP

limited number of other vehicles performing similar missions simultaneously

- General “ilities”
  - Safety - Ability to maintain control of vehicle in a failsafe manner with or without driver
  - Reliability – Ability to consistently produce the same results on operational hardware
  - Functionality - Ability to meet basic, quantifiable performance metrics and demonstrate performance via simulation
  - Scalability - Ability to partition the design into well defined, functional blocks with defined interfaces to allow parallel development
  - Testability – Ability to verify and validate system performance on the actual vehicle.

Overall system architecture was settled upon fairly rapidly by the design team and is illustrated in the following figure.

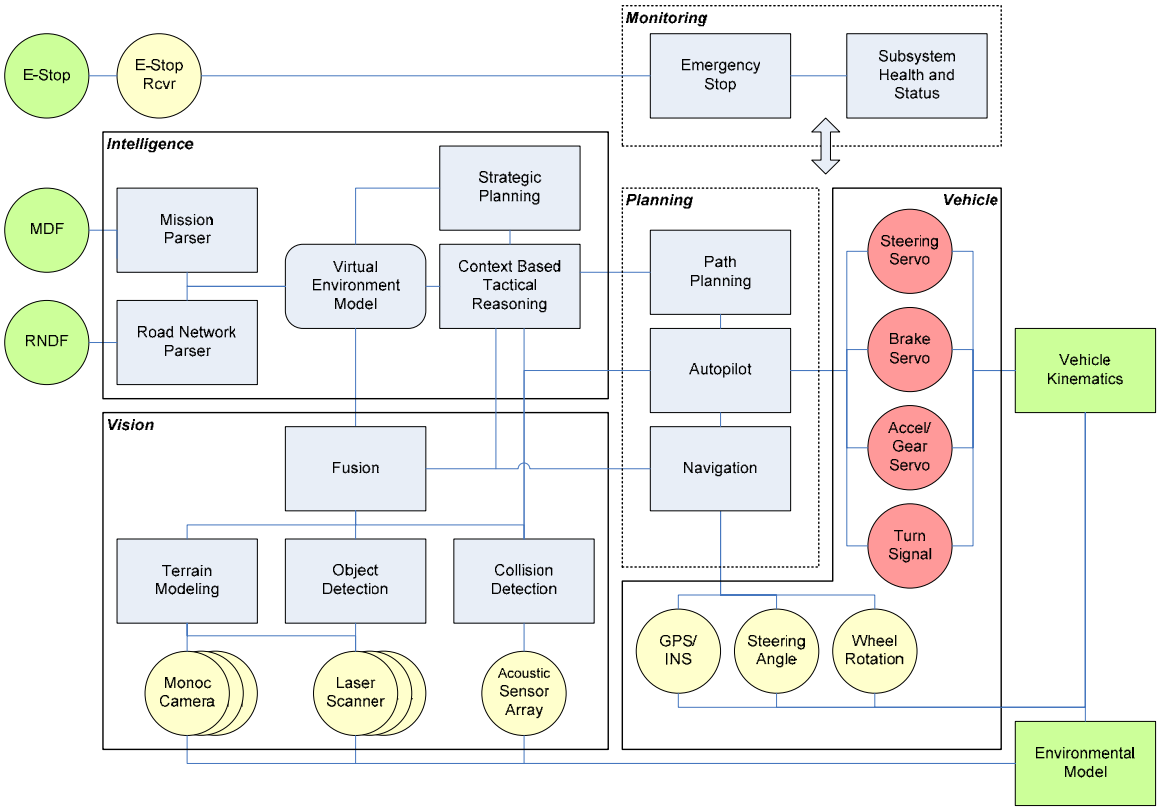


Figure 2. Functional Block Diagram

As a general rule, vehicle modifications were kept to a minimum. Actuators were designed to control existing vehicle hardware in a manner analogous to a human operator. For example, the steering servo, mounted along the vehicle centerline, controls the steering wheel with a belt system similar to the way a driver would control that system. This system easily allows both autonomous vehicle operation as well as driver operation. Servo torques and belt slippage are

adjusted to allow driver override even in the event of full system failure. Throttle and brake actuators are mounted under the driver and passenger seats and similarly provide failsafe operation. In particular, the brake actuator causes the vehicle brake to be depressed and is in the "always on" position via a spring mechanism. The brake is "released" via a pneumatic actuator that, should it fail, will cause the brake to return to the depressed position. Turn signal integration is accomplished via the vehicle's existing wiring infrastructure.

Own-state estimation (position, speed, heading as well as full vehicle attitude) is provided by an integrated GPS/INS from Oxford Technical Solutions. Rather than develop standalone Kalman navigation filters, the team opted for commercial hardware/software solutions. The Real-Time RT3000 family of products from Oxford provide very high-precision measurement of position, velocity, orientation, acceleration and angular rates for a wide range of applications. Inertial Navigation Systems combined with GPS provide high quality measurements, including obscured-sky speed measurement, in environments where the GPS alone struggles; lateral acceleration in a horizontal direction without the need to zero the accelerometer; and roll/pitch measurements which are accurate during continuous turns. The ability to integrate with wheel speed sensors provides sufficiently tight integration between navigation subsystem and the vehicle.

Vision systems are divided into three general categories, and while data is fused from all vision systems to present a coherent view of the environment, each category is primarily responsible for a single mission. SICK laser scanners mounted on a forward and rear mounting bracket, and rotating laser scanners mounted to the top vehicle rack, provide range and angle information to obstacles as small as traffic cones. The sensors provide information only for the leading edge of obstacles, but after multiple looks from varying angles obstacle geometry is refined. Scanner pointing direction and type were selected to optimize forward sector coverage. This approach also provides 100% overlap in coverage directly in front of the vehicle.

Visual cameras from Sony, essentially image-stabilized camcorders, are mounted to the roof rack and provide forward and rear views covering approximately the same viewing area as the laser scanners. The primary mission of these cameras is to provide lane detection from road markings and/or road edge boundaries. This information is used to augment a-priori information provided in the RNDF to more precisely define the road network to be traversed.

To augment vehicle safety, a series of ultra-sonic acoustic sensors provides near-field obstacle detection around the entire vehicle. Sensor range is limited to approximately 3m, and therefore in addition to basic collision detection, this sensor system essentially offers relative distance feedback when traveling at low speed so that precision maneuvers can be performed more accurately.

Processing is provided by an array of Linux and WinTel computers located in a shock mounted frame in the passenger's seat. Each computer controls one or more sensor/actuator systems and is responsible for primary data reduction and conditioning. Real-time operating systems provide that capability when necessary (any system operating at 10 Hz or greater). Computers communicate over a local Ethernet network, and various processes establish connection with one another, in a broadcast/subscribe manner, independent of their actual physical processor location.

Communication utilizes the Internet Communications Engine (ICE) framework which is a simplified derivative of the COBRA architecture. Overall health and vehicle status is monitored onboard, and in the event of sufficiently anomalous behavior the vehicle is commanded to a graceful stop.

Power distribution to all systems is provided by an array of DC-DC converters providing 5, 12, 24, and 48 volts throughout the vehicle. Power stabilization is provided by an array of deep cycle marine batteries which are recharged by the vehicle alternator.

Decomposition of the core software elements into Vision, Intelligence, and Planning elements is clearly visible in Figure 2 as are the key functional elements within these blocks and their interconnections. The detailed interfaces associated with health and status monitoring elements and e-stop are not shown for clarity. This diagram depicts the vehicle's functional block diagram and is identical to the function block diagram of a vehicle simulation which was constructed in parallel. Parallel system and simulation paths sped development and testing of system software. The use of ICE interfaces allowed vehicle and simulation to interconnect at various levels essentially proving a simplified hardware-in-the-loop capability.

The control system software interfaces with the vehicle through the actuators and sensors. Actuator control is handled from a single source, a low-level vehicle autopilot, while sensor inputs are divided into three basic groups: 1) vision-related, 2) navigation related, and 3) overall command and control (i.e., e-stop).

Overall mission inputs are provided by the RNDF, providing an initial seed of the system's environmental model, and an MDF defining the overall mission objectives in terms of waypoints and speed constraints. The overall system output is vehicle motion.

Viewed as a control system, the elements can be considered as follows: 1) intelligence develops a mission as a set of ordered goals to be achieved, 2) path planning efficiently plans a legal and drivable path to meet those goals, 3) the autopilot maintains the vehicle on path and within performance limits, and 4) PID controllers command various actuators to meet autopilot commands. Although difficult to see in the diagram, feedback effectively consists of four nested loops. The innermost loop consists of PID controller feedback (actuator position, etc.). The next loop consists of navigation information (position, speed, heading, etc.) used by the autopilot to develop control commands to maintain the vehicle on course. Beyond this is a path planning loop which effectively manages the tactical path based on tactical goals, bounds, and obstacles. At the outermost level is the overall intelligence loop that monitors whether or not the vehicle has met its tactical and strategic objectives. This outer loop is closed through vision as well as navigation, with tactical collision detection information interfacing directly with the autopilot.

System operation is straightforward. After a boot-up, self-test, the vehicle sits in a wait state ready to accept an RNDF and corresponding MDF. Once files are loaded and successfully processed, the vehicle remains waiting until released to execute the mission. The detailed mission plan is generated dynamically as the operational environment is discovered. Data logging is performed allowing mission playback for analysis.

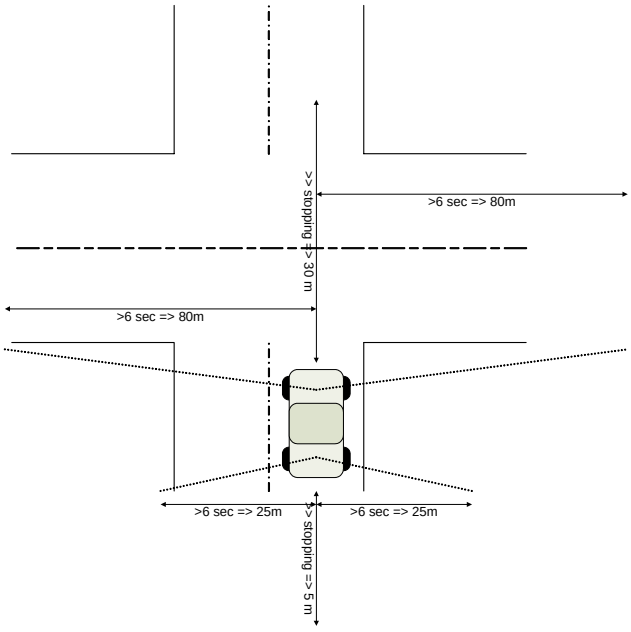
3. ANALYSIS AND DESIGN

It is perhaps fortunate that when preparing for the previous Grand Challenge, the team realized that neither a modified military vehicle nor a self-constructed platform was necessary to fulfill the mission requirements. The Subaru Outback Legacy falls more in the category of a "family car", yet its 4WD and off-road capabilities make it an excellent autonomous car. It's relatively small size, yet reasonably large and accessible interior space makes it an ideal choice for the Urban Challenge vehicle as well. Because of these previous decisions, the basic vehicle selection, the selection of actuators, and the conceptual design of vehicle sensors and software were effectively already made prior to the start of the Urban Challenge effort.

Of course the previous vehicle did not initially meet all the performance requirements of the Urban Challenge. From the basic requirements identified in Section 2, performance parameters, such as control loop update rates and sensor ranges were derived and identified as design-to goals early in the design process. Rather than use a requirements-based design methodology, however, a capabilities-based approach was employed. As long as delivered capabilities were at least consistent with design-to goals, systems were judged to have met objectives. As a result of these criteria and this approach, after initial design-to goals were established, the selection of specific approaches used on the vehicle was largely qualitative relying extensively on previous experience with the Grand Challenge. There were few true trade studies performed.

3.1. Vision

Vision subsystem design-to goals were largely driven by the challenging problem of crossing obstacles traveling at near full speed and/or stationary in lane obstacles. While certainly not all of the requirements, several key system requirements are summarized in the following figure.



- Near 360° obstacles detection
  - Stationary 0.5m diameter, 1 m high at nominal full-speed stopping distance (~30m)
  - Moving car-like in crossing scenarios with time to merge (6s … 80m)
  - Update rates > 2Hz
  - Rear obstacle detection at slow speed, no merge
- Lane boundaries
  - Define to nominal full-speed stopping distance (~30m)
  - 1m accuracy at range, 0.1m near-field

Figure 3. Nominal Vision Design-to Goals

3.1.1. Far-Field Obstacle Detection

One key requirement of urban navigation is the safe navigation in diverse traffic situations. With an allowed maximum speed of 13.5 m/s (30 mps) it is necessary to detect oncoming traffic and road-blocking obstacles in distances that are challenging for passive-only computer vision approaches. Laser Scanners that employ emitted laser light and the time-of-flight principle to deduce distances are an important instrument in this scenario, because their maximum range coincides with the desired design goal of about 80 meters. 2D Laser Scanners that use a rotating mirror to provide angular distance measurements in a plane are relatively inexpensive and widely available, but output only sparse information about the environment.

Nonetheless, a tactical placement of 4 laser scanners around the car allows for a near 360° field-of-view to detect static and dynamic obstacles in all possible locations relative to the car. The placement of these SICK LMS 291 scanners is indicated by red squares in Figure 4.

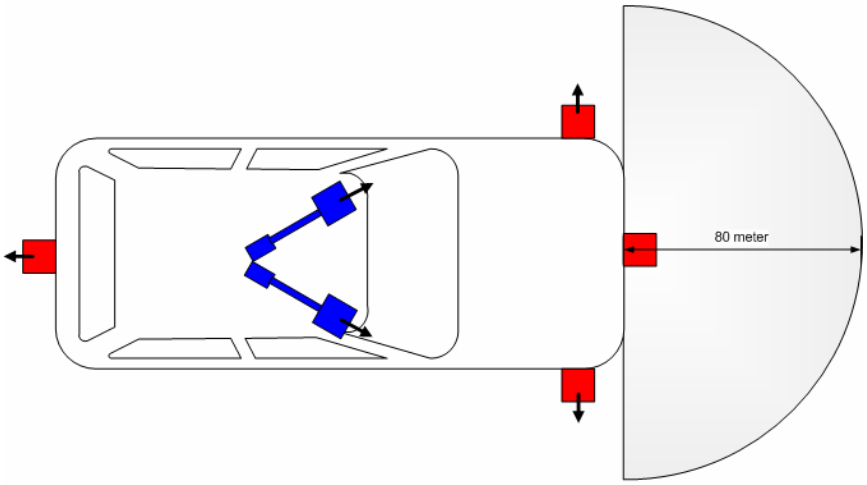


Figure 4. The placement of the laser scanners with arrows indicating the scanning direction. The 2D laser scanners that are mounted parallel to the ground are shown in red. The rotating 2D laser scanners are shown in blue. The scanning angle and range is exemplary shown for the front scanner.

For most situations that will be encountered in the Urban Challenge, the lasers scanning parallel to the ground will be sufficient, but any height information of obstacles gets lost and no scan points on the driven ground plane can be provided. This is undesirable since the height of obstacles can be used for classification of objects but perhaps more importantly height information significantly augments terrain (i.e., road) identification and following performance.

These reflections led to the conclusion that a 3D scanning scheme needs to be designed. Commercial 3D laser scanner systems that scan a predefined volume are available, but very expensive and the scanning operation is very time-intensive. TeamUCF has devised a novel and efficient method to employ conventional 2D Laser Scanners to generate 3D models of the environment. In collaboration with the Mechanical Engineering department, we developed an actuated mount that rotates the 2D scanners to generate 3D samples. Using those 3D points it is possible to locate and classify obstacles in each scenario encountered in the Urban Challenge. See the blue squares in Figure 4 for the scanner locations.

A computer-generated model of the actual mounted scanners and a simulated scan in an urban environment is presented in Figure 5. The resulting scan is rich in detail and provides a natural 3D representation of the environment. As of the publication of this report, 2D laser scanning systems are installed and tested and 3D scanners are in the process of being installed.

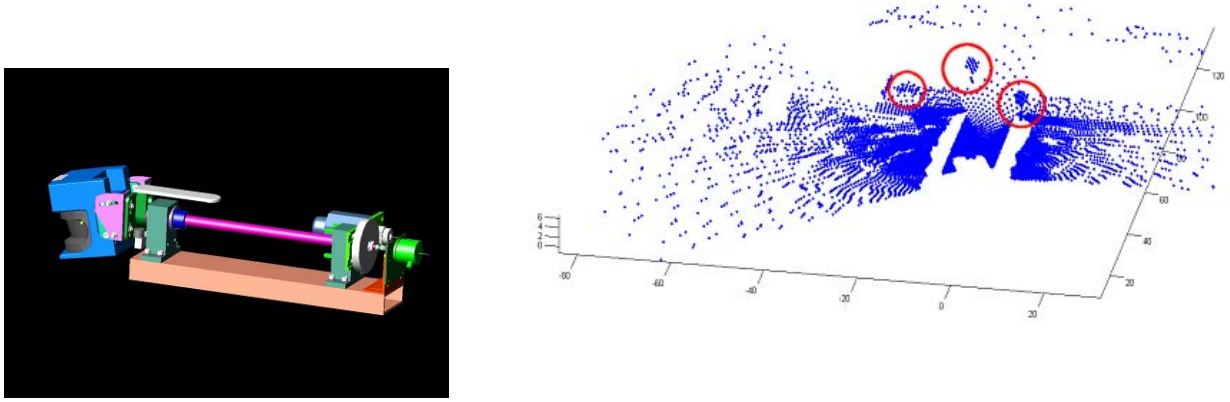


Figure 5. (Left) 3D Model of one of the rotating 2D Laser Scanners. (Right) A simulated scan of an intersection with detected stop signs and another car marked with red circles.

3.1.2. Near-Field Collision Detection

Figure 6. One of the 8 ultrasonic arrays.

For near-field collision detection ($\leq$ 3 meter), the laser scanners output unreliable information due to a minimum scanning distance of about 2 meters. The laser scanners are augmented by a battery of short-range sensors. TeamUCF made the design decision for 8 arrays of ultrasonic sensors. Each of these arrays contains 5 Devanech acoustic sensors that are positioned to allow coverage of 180°, see Figure 6.

The 8 arrays are connected to a central Rabbit microcontroller that can query the sensors in different modes. Multiple working modes that can serialize and interleave the sensor polling sequences reduce the negative effects of crosstalk. Individual sensor can be queried at any time to resolve ambiguities in detection.

The 8 mounts are places around the car body and placed above or below the laser scanners if feasible. For an illustration of the placement see Figure 7.

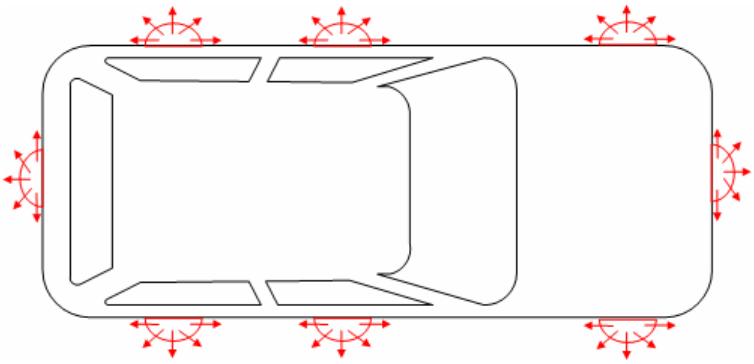


Figure 7. The placement of the near-field ultrasonic arrays around the car. This provides a complete coverage around the car for distances $\leq$ 3 m.

3.1.3. Vision-Based Lane Detection

In order to safely navigate the DARPA Urban Challenge course in the absence of dense waypoint information, the vehicle is equipped with a Vision-Based Lane Detection System. This system is capable of providing road identification and lane detection sufficient to guide the vehicle without additional sensor input.

For the past two years, the ground vehicle team for the Robotics Club at UCF has performed lane line identification on different terrain for the Intelligent Ground Vehicle Competition (IGVC) sponsored by the Association of Unmanned Vehicle Systems International (AUVSI). At the IGVC, the autonomous vehicles are required to identify and follow a lane cluttered with different types of obstacles, terrain, and lighting conditions using machine vision. High success rates have been achieved using different filtering processes combined with the Hough Transform to identify white lane lines. The primary system input is a Sony HDR-HC3 camcorder mounted to approximate the driver point of view. The camcorder is connected to the Vision Processing Unit via Firewire (IEEE1394) interface. Images are sent to the VPU at a frame rate of 30Hz. The VPU outputs lane information in real-world coordinates at the rate of 10Hz exceeding design-to goals.

Lanes are detected using a correlation based approach using a typical center line marking as the primary feature to discover. This is modeled primarily as a correlation problem using a modified version of the U and V channels of a YUV image to minimize sensitivity to variations in intensity. The algorithm operates as follows:

- Calculate modified UV channel image at each pixel.
- Downsample modified UV image as necessary to meet performance requirements.
- Run Zero-Normalized Cross-Correlation between example patch and image.
- Threshold ZNCC image.
- Perform Probabilistic Hough Transform on thresholded ZNCC image.
- Threshold Hough Transform results
- Extract segments from results
- Identify boundaries from segments (left, right, center)

Figure 8 shows an example input frame. Figure 9 gives the corresponding output frame from the lane detection system. The white lines in the input image indicate the lane boundary as discovered by the vision algorithm. In addition, there are several "grayed out" areas in the output image. These are candidate lane markings. Candidate lane markings are areas which had a high level of response to the 16x16 patch but were rejected as being unsuitable. Currently, we are able to successfully guide the vehicle using only camera based vision interfaced to the AI/Path Planner at speeds below the desired maximum operational speed of 13.5 m/s (30mph). This has lead to operation of the vehicle in a "lane discovery" context when encountering roadways for the first time.

Figure 8. Input Image

Figure 9. Output Image

3.1.4. Sensor Fusion

The sensor fusion module's task is to gather information from the far-field and near-field sensors and combine them in a meaningful way to provide an environment model to Intelligence. For this purpose a 3D probabilistic occupancy grid is built by projecting the pose-corrected scanner readings into a 3D map. This map is comprised of 3D voxels with a volume of 0.5 x 0.5 x 0.5 m^3. Each voxel takes on a binary value of either "occupied" or "free". Due to the probabilistic nature of this approach, mis-readings from sensors cannot accumulate enough hits in a voxel to turn its value to occupied, but spatially consistent readings of obstacles will result in fast detection.

All voxels posing a danger to the car are extracted and condensed in a 2D occupancy map. By using the techniques of Connected Components and Polygon Simplification, obstacles are published as collection of concave polygons to the Intelligence module. Each obstacle is tracked by an Extended Kalman Filter that provides a robust tracking for static and dynamic obstacles and attaches the respective trajectory information to the publication.

The output of sensor fusion after one complete round on the site visit course is shown in Figure 10. The outline of the submitted course is shown with dark blue lines and the detected obstacles with light blue polygons.

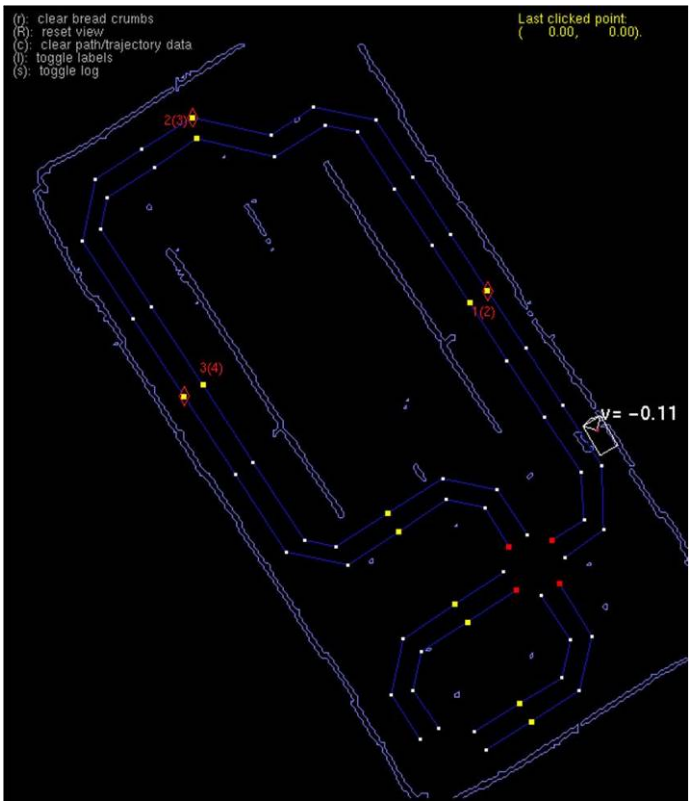


Figure 10. Environmental Model Information after Course Driven Once.

3.2. Intelligence

Design-to goals for vehicle intelligence subsystems include primarily qualitative objectives that can be logically inferred from the design partitioning already described. Update rate requirements are driven largely by vision capabilities and general requirements associated with vehicle motion at operating speeds and were selected to be 2 Hz.

The approach employed utilizes technology developed at UCF for real-time control of tactical agents known as Context-based Reasoning (CxBR). It provides an organization for all intelligent elements utilized in the vehicle, while at the same time permitting modules contributed by other researchers to be seamlessly integrated within the architecture. CxBR is a human behavior representation paradigm successfully used to model tactical human behavior in simulations. It is based on the concept that humans think in terms of contexts. That is, we recognize the context we are in and apply a predetermined set of behaviors (actions, functions, rules, etc.) that permits us to manage our task when in that context.

CxBR also provides for situational awareness by recognizing when the context has changed, as it almost always repeatedly does in certain as well as uncertain environments. Upon detection of a changing situation, CxBR decides which new (but pre-existing) set of behaviors should become "activated" and thus provide the appropriate behaviors for the newly emerging context.

Controlling an autonomous entity in a tactical mission involves a continuous sequence of contexts over time, to which CxBR applies a sequence of predetermined but flexible behaviors to address the current situation. These modules of context-sensitive behaviors are referred to generically as contexts. Within this term, we have a hierarchical organization of such contexts. At the highest level is the mission context, where the mission or task is defined. A mission context does not control anything, but does provide critical information about the upcoming task or mission. The main element of CxBR is the major context. Major contexts are the main control element for the agent. They contain two major components: 1) The behaviors appropriate for that context modeled in the form of C++ functions, and 2) the knowledge required to recognize when that major context is no longer relevant and should be deactivated in favor of a relevant one. The last element, or actually, class of elements, is the minor context. Minor contexts are similar to major contexts, except that they work within upper-level contexts. Furthermore, unlike major contexts that occupy only one level in the hierarchy, there are an arbitrary number of minor context levels. For example, there are sub-contexts, sub-sub-contexts, sub-sub-sub-contexts, and so forth. When a minor context completes its work or is no longer relevant, control goes back to its immediately upper-level context ion much the same way that a called function returns control to its calling function. Figure 11 is illustrative of a collection of sub-contexts associated with the context of lane driving. Key information required in the context and context behavior, in terms of sub goals is apparent. Other contexts include lane discovery, zone driving and intersection driving.

To ensure consistency in the communication to the path planner, each context provides a method that return speculative car guidance even before the conditions for that context are met. This allows another context to augment its guidance by speculating the next context and using the other context's guidance after the current. When the context switch actually occurs, and provided that conditions have not changed, the car guidance remains continuous. For example, the lane following context may consult the intersection context in order to extend car guidance before arriving in front of a stop sign. When the intersection context takes control, the car is already slowing down thus preventing a sudden change in guidance that would otherwise occur.

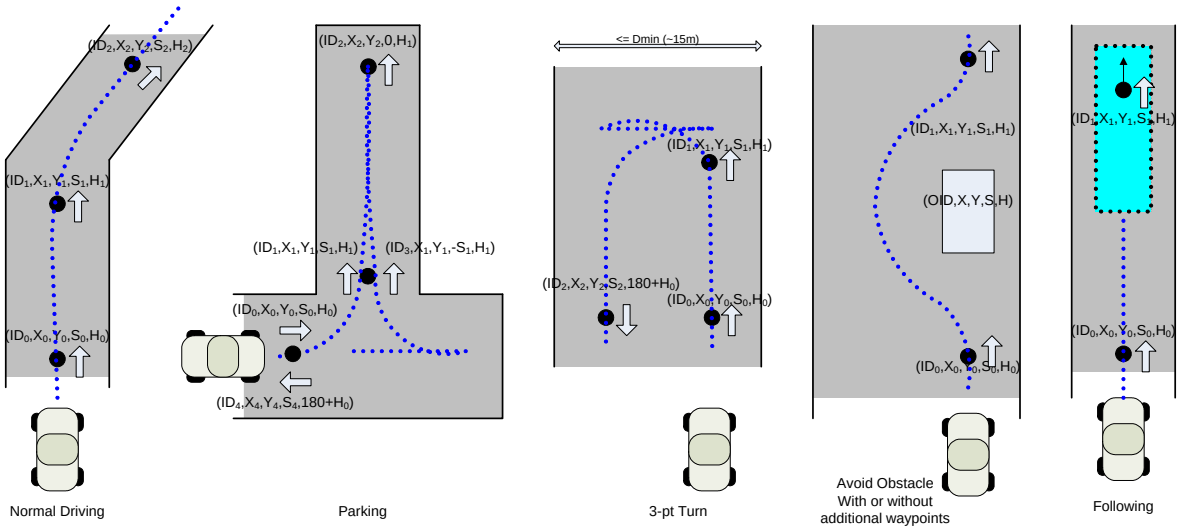


Figure 11. Illustrative Example of CxBR Contexts

Each context is responsible for providing guidance to the path planner. In doing so, each context filters the information provided by the sensor system to better guide the path planner operation.

Route planning is implemented by converting the road network into a directed graph. Lane to lane or lane to zone transitions are represented by nodes in the graph. Traveling on a lane is represented by edges. In addition, nodes are used to represent parking, U-turns and lane changes. The graph generation algorithm creates the graph by first considering the road network as specified in the RNDF. Lane changes are added between appropriate nodes. U-turn nodes are added as a speculation during minimum path generation. The route planner uses a dynamic weight assignment algorithms that assigns a cost to each node and edge. A traditional minimum path algorithm is used to find the best path. The route planner uses the current position of the car and the location of the next checkpoint to generate four possible routes, enumerating the situations where a U-Turn is or is not performed at the start and right before the end of the path. The shortest of these routes is selected and the AI context/state is then responsible for following the route. The route planner is running every few seconds, each time with updated weights for each edge and node. This dynamic reassignment of weights is used to re-assess a current route. For example, the weight of crossing a lane starts with a fixed cost that depends on the known or estimated length, but then grows with time if progress is not made while traveling along it. That way, if the car gets stuck in a traffic jam, the cost of following that road will eventually exceed the cost of a U-Turn followed by the other route.

A direct result of separating vehicle control in this manner is that the CxBR intelligence system must monitor vehicle performance independently of low level control systems. This is analogous to the operation of a back-seat driver providing directions to the actual driver of a vehicle. The two “controllers” may independently assess whether or not goals are being met. The level of fidelity of specific objectives largely determines the control, or micro-management, of the specific mission. It is felt that this added complexity of monitoring is outweighed by the benefit of independent, parallel system development and the ability to apply the CxBR engine to a wide array of control problems with very little modification.

3.3. Planning

System requirements decomposed to design-to goals for the planning subsystem that effectively partitioned planning into three levels of control. Safety testing is performed at every level of control. For example, throttle activation is disabled when the vehicle is in neutral and vehicle steering angles are restricted by vehicle lateral acceleration limits independent of the implied commands provided to these low level functions.

The bandwidth requirements for hardware control led to the use of independent PID controllers for steering, throttle, and brake control, where individual PID parameters were tuned for the specific performance of the actuator being controlled.

Vehicle speed and steering angles are controlled via an intermediate level vehicle autopilot consisting most importantly of a PID control loop with vehicle position and heading feedback driving vehicle steering angle. The fundamental objective of the loop is to drive the vehicle along a dense path with reasonable error. Given the nature of the missions, a maximum error tolerance of 0.5m is used as a guiding parameter for most situations with a nominal response

time of several seconds for large motions and tenths of seconds for small motion. Autopilot operation is performed at 10 Hz to meet these objectives. Nominal operation associated with an obstacle avoidance maneuver is illustrated in Figure 12.

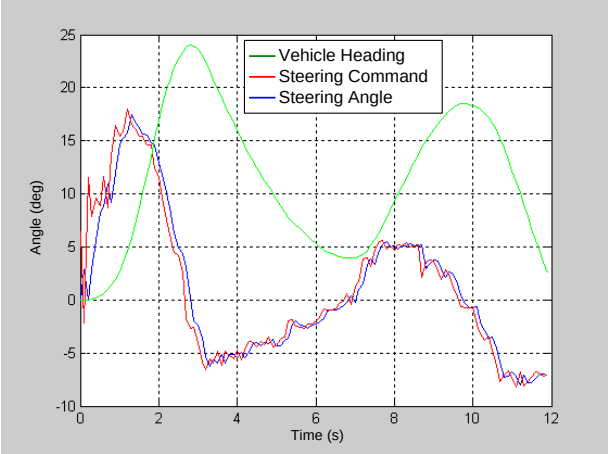


Figure 12. Typical Autopilot Performance

The most interesting element of planning is perhaps the generation of dense path information from the relatively sparse information provided by Intelligence. The fundamental objective of this element is for a real-time path planning and control algorithm by which the vehicle can autonomously navigate through a dynamically changing and congested environment. In the urban scenarios the vehicle must operation within very tight space, must avoid static as well as moving obstacles, and must perform complicated maneuvers. To accomplish this, a two-component path planning algorithm was implemented:

- A quick search algorithm for a few waypoints within the maneuverable space of the environment;
- A real-time polynomial-parameterized path that ensures maneuverability (by satisfying the vehicle's kinematic constraints), smoothness, and efficiency of the path.

The former is well known and widely used, and the latter has an analytical, optimized solution (and hence computationally efficient) and also renders the corresponding control commands. In what follows, the second component is outlined.

To ensure that a path planned is realized by the vehicle and also collision-free, vehicle kinematics and physical envelope must be considered. Illustrated in Figure 13, the Knight Rider vehicle has a rectangular envelope, and its kinematic model is that of a front-driving and front-steering vehicle, that is,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \sin\phi / l \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2, \qquad (1)$$

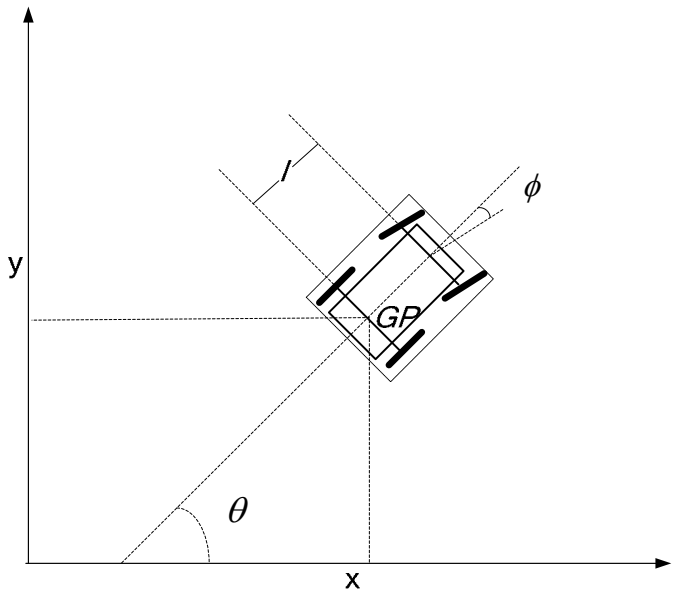


Figure 13. Kinematics of Front-Drive / Front-Steering Vehicle

where *x*, *y* are the Cartesian coordinates of the rear wheel, θ is the orientation of the vehicle body with respect to the *x* axis, and ϕ is the steering angle, v_1 is the driving velocity referring to the front wheel, and v_2 the steering velocity. It is well known that, under the following change of coordinates and input transformation

$$x_1 = x, \quad x_2 = \frac{\tan\phi}{l\cos^3\theta}, \quad x_3 = \tan\theta, \quad x_4 = y, \tag{2}$$

and

$$u_1 = v_1\cos\theta\cos\phi, \quad u_2 = \frac{v_2}{l\cos^3\theta\cos^2\phi} + \frac{3\sin\theta\sin\phi}{l^2\cos^4\theta}v_1, \tag{3}$$

the vehicle's kinematic model can be transformed into the canonical chained form:

$$\dot{x}_1 = u_1, \quad \dot{x}_2 = u_2, \quad \dot{x}_3 = x_2u_1, \quad \dot{x}_4 = x_3u_1. \tag{4}$$

Thus, a collision-free path properly selected from the class of 6-th order polynomial solutions to chained form (4) is guaranteed to be maneuverable. Given the starting position and the destination in the environment, a few waypoints can easily be found to generate a rough route of straight line segments, hence the complete mission is segmented into a series of subtasks each of which admits many 6-th order polynomial solutions to (4). Thus, the real-time path planning and control boils down to solve the optimal planning problem for each subtask.

Suppose that the subtask is to make the vehicle move from the waypoint *A* to the waypoint *B*. The corresponding motion planning problem for vehicle (1) becomes to find the control inputs v_1 and v_2 to steer the vehicle from the initial configuration $q(t_A)$ to the final configuration $q(t_B)$ while minimizing a suitable performance index, where $q = [x, y, \theta, \phi]^T$, t_A is the starting time instant for subtask and t_B the ending time instant. The main features of this path planning approach are that the solution is analytical and kinematic constraints are explicitly considered while addressing the problems of avoiding the moving obstacles and generating optimal

trajectories. The algorithm is particularly favorable for the mission of vehicle docking and parking in a loaded area.

In essence, the trajectory from the waypoint *A* to the waypoint *B* can be parameterized using a 6 order piecewise-constant polynomial as

$$y = a_0^k + a_1^k x + a_2^k x^2 + a_3^k x^3 + a_4^k x^4 + a_5^k x^5 + a_6^k x^6, \quad (5)$$

where $a_i^k, i = 0,...,6$ are the coefficients to be determined based on the boundary conditions $q(t_A)$ and $q(t_B)$, the obstacle avoidance criterion generated according to the environment, and the given optimal performance index. The changing environment (with moving obstacles) can be described by a sequence of piecewise-constant route conditions according to the sensor information. Though the motions of obstacles are not modeled or known a-priori or predictable, their geometrical shape (envelope) and the corresponding locations and motion velocities are all detectable as soon as they emerge within the sensing range of the vehicle. The envelope of obstacles can be mathematically modeled using a combination of simple arcs and line segments. Accordingly, the collision avoidance criterion is established by satisfying a set of simple inequalities which can guarantee that the distance between the vehicle and obstacle is greater than reasonable value for safety. Considering the complexity of environment and to improve the efficiency, the "hard" obstacles and "soft" obstacles are dealt with respectively. Among them, "hard" obstacles refer to those must be avoided, such as human beings, other vehicles in the environment, while "soft" obstacle is preferred to be avoided but can be run over/through by the vehicle, such as unpaved or gravel area, potholes, and "splash" and "spray" on the wet road generated from other vehicle's wheels.

The solution to (5) based on obstacle avoidance criterion renders a family of feasible collision-free trajectories. A L_2 norm type optimal performance index which measures the closeness of the feasible trajectories to straight line can be introduced for analytical solving the optimal value $a_i^k, i = 0,...,6$ to pick the optimal path from the obtained family of feasible collision-free trajectories. Once the optimal value of $a_i^k, i = 0,...,6$ is found, the corresponding steering control inputs v_1 and v_2 can be straightforwardly derived according to the chained system given in (4) and transformations (2) and (3). The overall effect is illustrated in the following figure.

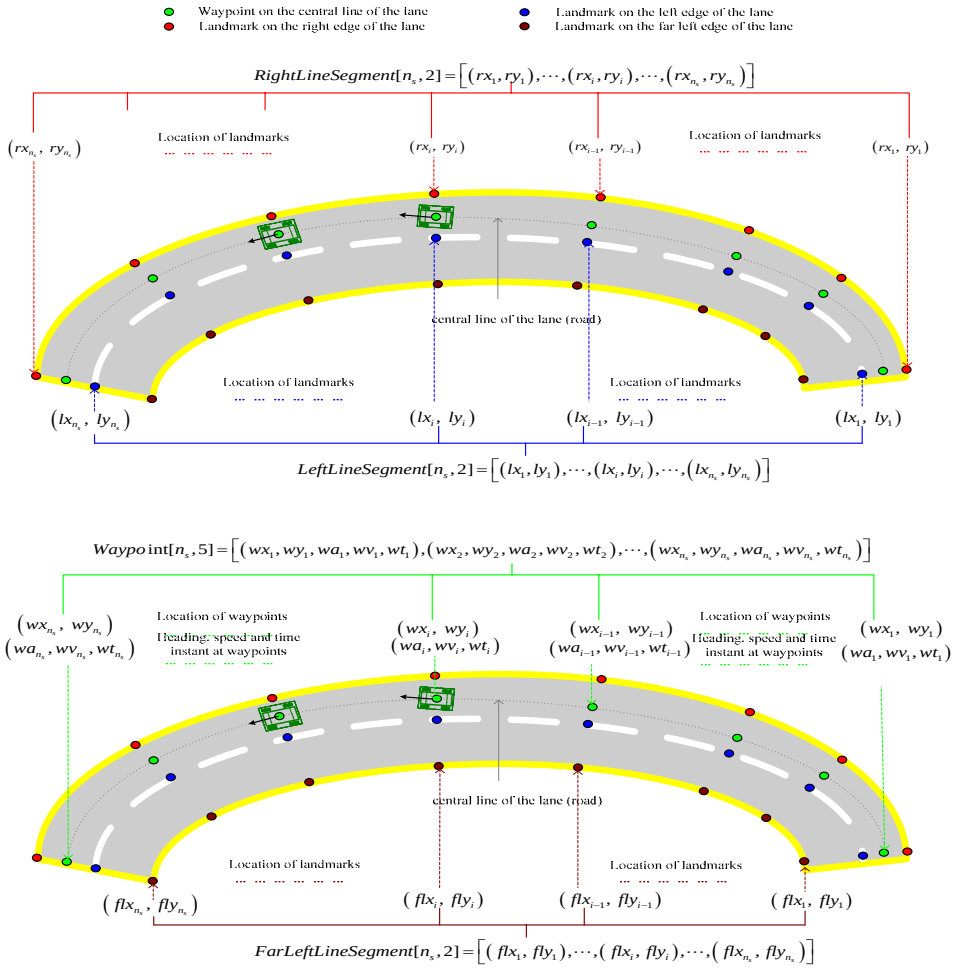


Figure 14. Mathematical Model of Path

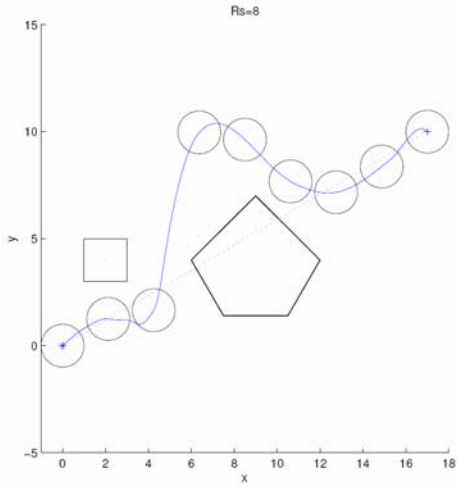


Figure 15. Path with Static Obstacles

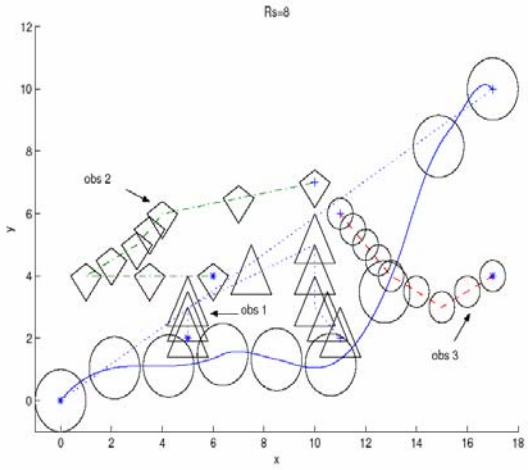


Figure 16. Path with Moving Obstacles

3.4. Other Subsystems

Knight Rider navigation fuses a number of sensors to provide an accurate determination of the current vehicle state which includes position, heading, speed, and attitude. Attitude information is used specifically by sensor subsystems to mad sensor relative geometry measurements into a world frame for inclusion in the environmental model. The vehicle's existing ABS sensors are used to provide the current speed of all four wheels and this information augments states maintained in the Oxford GPS/INS. Differential corrections are provided the GPS/INS. UCF has investigated dual antenna performance to augment attitude information, but performance was insufficiently different from the single antenna system now employed to warrant the complexity and idiosyncrasies of such a system. Position accuracy of such a system is << 10cm while angular accuracy is approximately 1°. The fusion of sensor data is provided within the GPS/INS and this device provides state estimation at a rate of a least 10Hz.

4. RESULTS AND PERFORMANCE

The overall performance of the vehicle can perhaps be best illustrated in a relatively simple driving scenario. Consider the mission illustrated in Figure 17. In this case the vehicle is located at the southern end of a parking deck on which an oval loop course has been identified via lane markings as well as a sparse set of waypoints. The green circles in the figure illustrate the checkpoints to be followed as defined in the MDF, while open circles are optional waypoints used primarily to define lanes. The Knight Rider plans a loop course initially unaware of the obstacle located on a checkpoint. The Knight Rider identifies the obstacle as it passes the third turn (the obstacle is initially shielded from view by walls at the center of the course). After the obstacle is detected a new route is identified which first brings the vehicle to a stop at a safe distance behind the obstacle and then includes a path around the obstacle, bypassing the checkpoint. Logged data shows vehicle speed and steering angle as a function of time over the course. Slower speed in turns and a pause prior to maneuver around the obstacle are both visible in the data.

For another view of obstacle detection, path planner, and navigation consider Figure 18 In this case, the vehicle avoids a static obstacle car that is located on the western straightaway of the course. Both figures illustrate an obstacle avoidance situation by showing the image recorded by the frontal camera on the left and the output of obstacle detection and path planning on the right side. In the right images, the vehicle is shown as white rectangle with its speed (m/s) attached. Knight Rider's driven path is marked in red, the detected obstacles as blue polygons, and the planned path as green line. Furthermore the zone with the static car obstacle is emphasized by a yellow dotted circle. The detection range in Figure 18 is approximately 50 meter and a smooth path around the obstacle is planned and executed by the autopilot.

Figure 17. Loop Course Performance

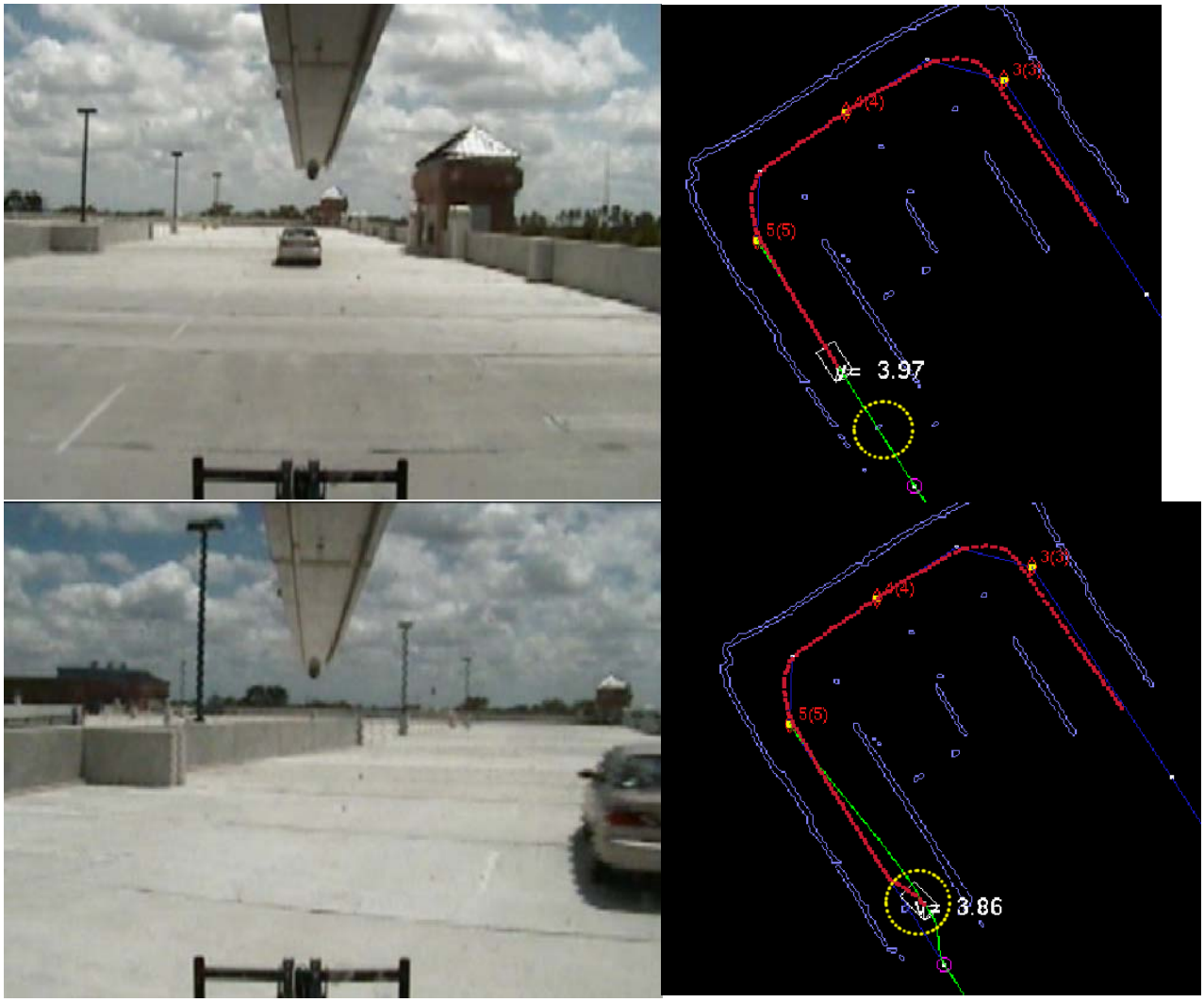


Figure 18. Detection of obstacle as viewed from onboard camera. Output represents the planned and executed path.

5. REFERENCES

[1] Team UCF. DARPA Grand Challenge 2005 - Technical Report, University of Central Florida, August 2005

[2] Eckerson, Wayne W. "Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications." `Open Information Systems 10,` 1 (January 1995): 3(20).

[3] Schussel, George. `Client/Server Past, Present, and Future` [online]. Formerly Available WWW <URL: http://news.dci.com/geos/dbsejava.htm> (1995).

[4] Brian P. Gerkey, Richard T. Vaughan, and Andrew Howard. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In Proc. Of the Intl. Conf. on Advanced Robotics (ICAR), pages 317–323, Coimbra, Portugal, July 2003.

[5] OMG. CORBA/IIOP Specification. Object Management Group, Inc., Needham, MA, 2001

[6] Douglas C. Schmidt. Patterns and Performance of Real-time Object Request Brokers. Distributed Object Computing Group, University of California, Irvine, 2000.
[7] Zhihua Qu, Jing Wang, and C. E. Plaisted, “A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles”, *IEEE Transactions on Robotics*, Vol. 20, pp 978-993, 2004.
[8] Jian Yang, Abdelhay Daoui, Zhihua Qu, Jing Wang, and Richard A. Hull, “An optimal and real-time solution to paramerterized mobile robot trajectories in the presence of moving obstacles”, 2005 *IEEE International Conference on Robotics and Automation*, pp. 4423-4428, Apr. 18-22, 2005, Barcelona, Spain.